Optimization of Continuous Problems with Evolutionary Algorithms

Dr. Ralf Schlatterbeck
Open Source Consulting

Email: office@runtux.com
Web: http://www.runtux.com
Tel. +43/650/621 40 17

Evolutionary Algorithmm

- Search for an optimal solution to a problem
- Inspired by nature
- Genes are sequences of numbers or symbols (Binary, Integer, Floating-Point)
- From a gene we build a potential solution to the problem: phenotype
- Algorithm works with a population of a specified size
- Fitness: Objective function is optimized
- Maximization or minimization is possible
- *Individual*: genotype + phenotype

Sketch of Algorithm

1. Random initialization of individuals
2. Evaluation of individuals (fitness)
3. Loop: Create next generation
   (a) Selection: Pick individuals for next step
   (b) Use genetic operators on selected individuals (“mixing”)
   (c) Evaluation of new individuals
   (d) Creation of next generation from new and old individuals (“replacement”)

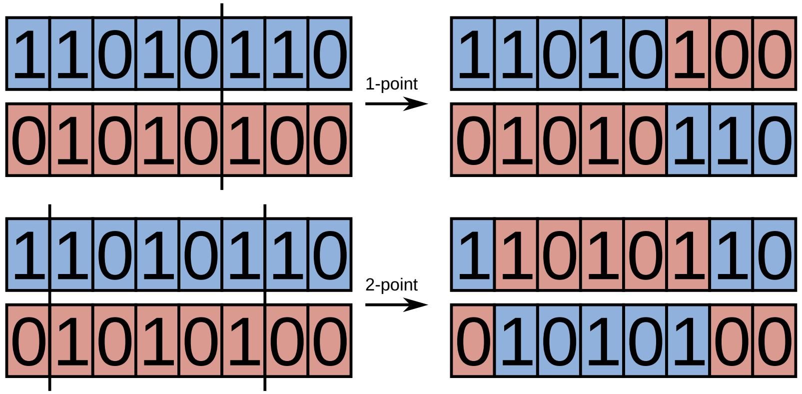
Selection pressure

Without selection pressure we have no search, only a random walk in the solution space – Variants:

- Selection in the first step of the loop in *(a) Selection*
- Selection in the last step of the loop in *(d) next generation*

Different variants of evolutionary algorithms have differing strategies

Crossover



Floating-point vs. classic

- Not enough “Mixing” with classic approach
- Crossover should use properties of both parents
- . . . without just copying them
- does not work well enough with floating-point: Not enough mixing

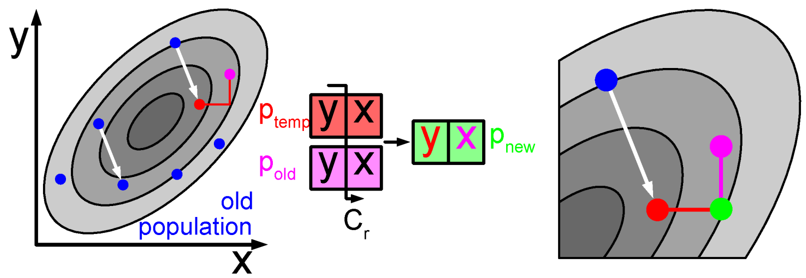
⟶ We need a different approach that mixes alleles

⟶ One of them is Differential Evolution

Differential Evolution

- Gene is a vector of floating point numbers
- For all genes:
- Add weighted *difference* of two genes to third gene
- Cross over with current gene
- Crossover uses at least one “allele” of the new gene
- Crossover probability depends on problem either small (0.2) or large $(0.9 - 1.0)$
- Example of two-dimensional case: (x, y)

Differential Evolution Crossover



Differential Evolution Crossover [Sch23]

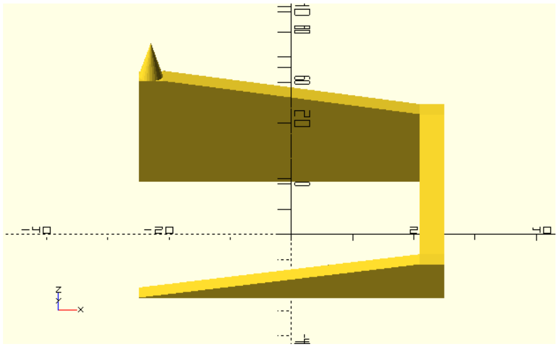
Algorithm Differential Evolution

Only for floating-point gene! [SP97, PSL05]

(a) Selection: Take each individual k in sequence
(b) Randomly select three individuals all different and different from k
(c) Compute weighted vector difference from two of these individuals and add to third
(d) Optional: Crossover of result with k
(e) Next generation: The new individual replaces the old one if it is better

Numeric optimization!

Differential Evolution Visualization



Parcours for Differential Evolution (OpenSCAD)

Implementation in PGApack

- PGApack: Parallel Genetic Algorithm Package
- Open Source genetic algorithm implementation
- Originally from David Levine, Argonne National Laboratory
- Parallel execution via MPI (Message Passing Interface)
- Construction kit for your experiments
- I'm maintaining a fork since 2017
- Current version implements differential evolution and other variants
- PGApy is a python-wrapper (since 2005)

Constraints

- Simple "‘box constraints"’
- Min and max of each parameter (=gene)
- e.g. with lengths: Never negative
- . . . and most often the order of magnitude is known (mm, m, oder km)
- Is supported by almost all evolutionary algorithms
- Complex constraints: Like additional objective functions
- Eval-function returns several values (some of them are constraints)

Several Objective Functions

- Traditional: Just one objective function
- Is minimized or maximized
- Multiple objective functions:
  - Either only for constraints
  - Or for multiobjective optimization

→ Constraints make the optimization problem more difficult

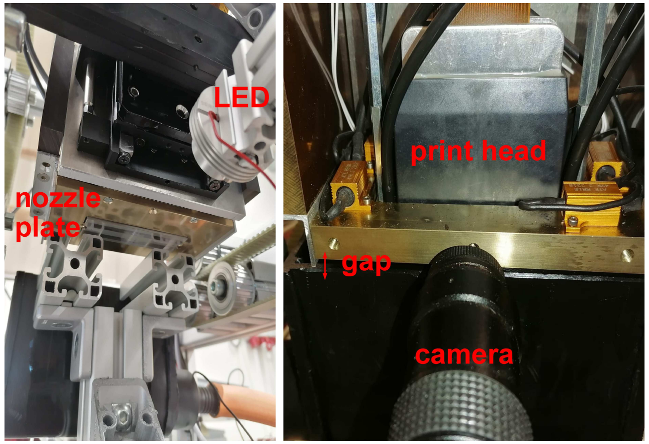
→ Multiple objectives can contradict each other!

→ Pareto-optimization

Waveform Design for Inkjet

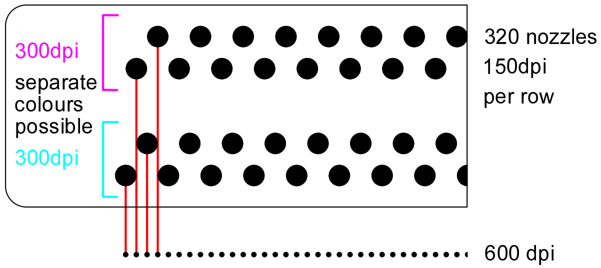
- Piezo-Electric Inkjet
- Nozzle width 23µm
- Droplet size around 6pL (picoliter)
- Waveform determines droplet quality
- Print quality: Speed, satellites (small drops after main drop), droplet size, stable at high jet-rate, . . .
- Analytic physical/mathematical solution is complex with many simplifications (to be able to compute it at all)
- Printhead dimensions are not given by manufacturer

Dropwatcher



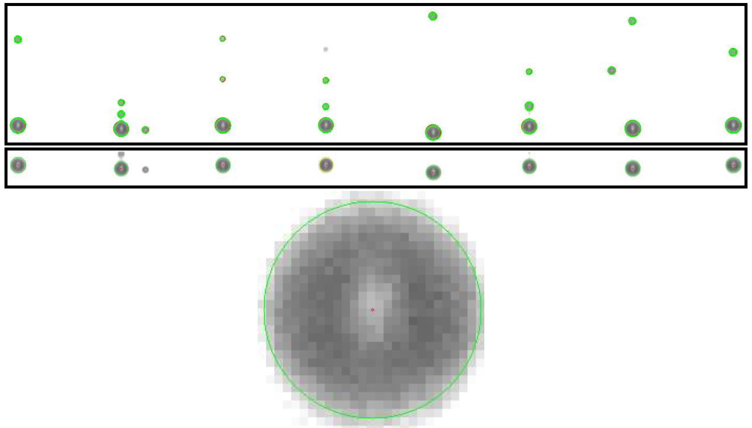
Dropwatcher Hardware [Sch23]

Print Head



Print Head [Sch23]

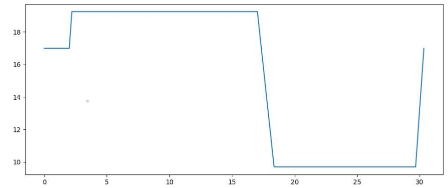
Image Recognition



Blob- and Circle Detection [Sch23]

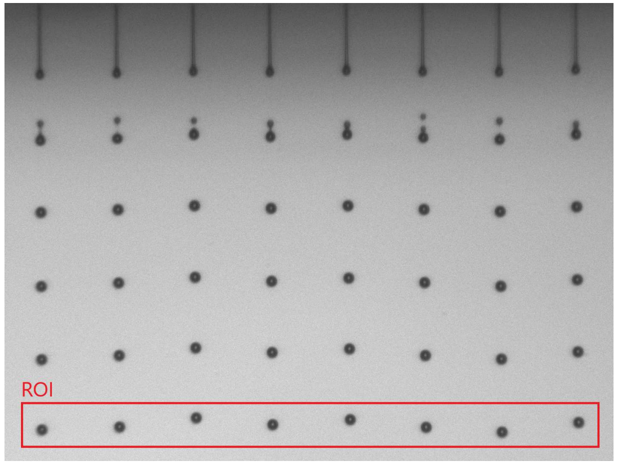
Optimization with Differential Evolution

- Waveform parameters set by genetic algorithm
- Fitness determined from image recognition
- e.g. minimum speed, size of drop, number of satellites . . .



Typical waveform [Sch23]

Drop Fronts



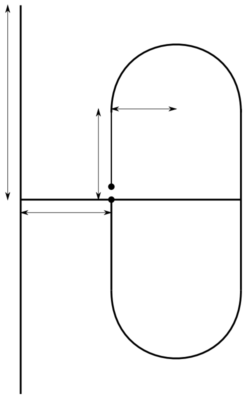
Drop Fronts Jetting at 32kHz [Sch23]

Example: Antenna

- Simulation with numerical electromagnetics code
- NEC was written in the 1980s using punched cards
- Rewrite in C/C++/Python `nec2c`, `necpp`, `PyNEC`
- nec2c: Translate NEC input file into lots of tables
- xnec2: Inputs a NEC file and shows graphics
- xnecview uses nec2c output and shows graphics
- . . . all packed in Debian
- My own plot-antenna can also show graphics

Antenna Optimization using genetic algorithms was published in the 1990s [Lin97] but hasn't seen much use, maybe because it was patented until 2015? [AL95]

2-Element Antenna with Folded Dipole



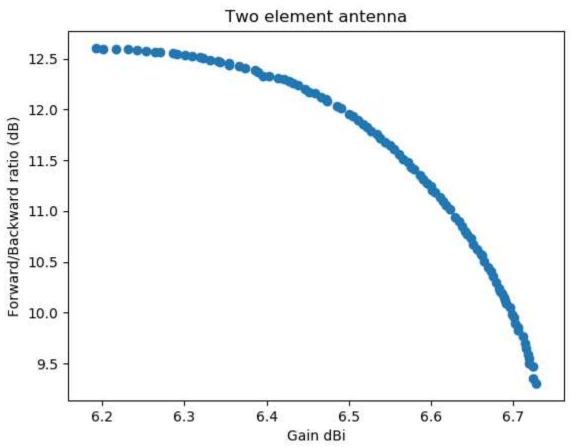
- Four parameters
- Coding as floating-point

$\rightarrow$ Differential Evolution

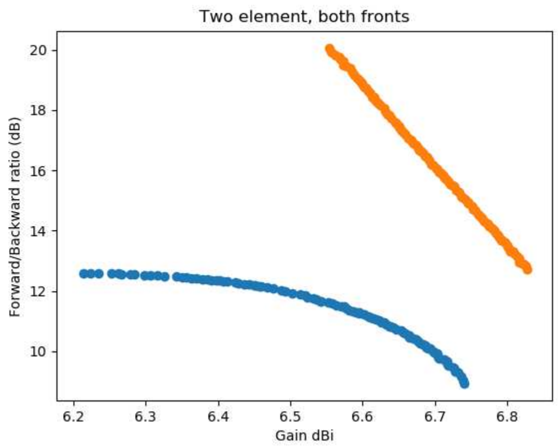
Antenna-Optimization with NSGA-II

- Multi-objective optimization with “Non Dominated Sorting Genetic Algorithm” NSGA-II
- One Constraint: VSWR < 1.8
- Two objectives: Gain and front-to-back ratio
- Gene of length 4:
- Length of folded dipole
- Width of folded dipole
- Length of second element
- Distance of elements
- Box constraints for all parameters

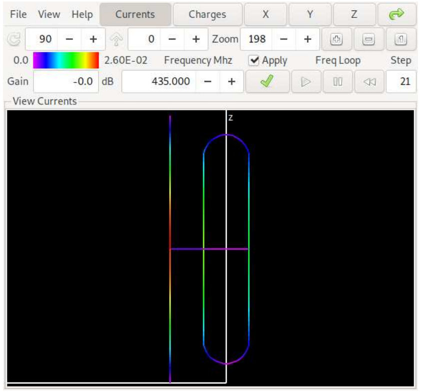
Pareto Front of Optimized Antenna

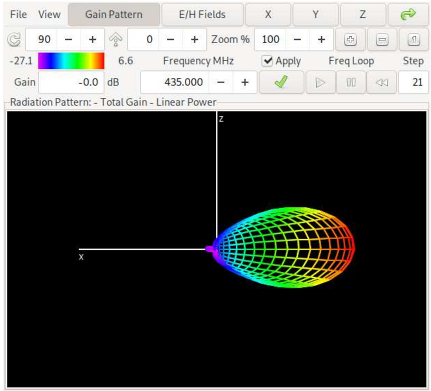


Pareto Front for Different Optimization Run

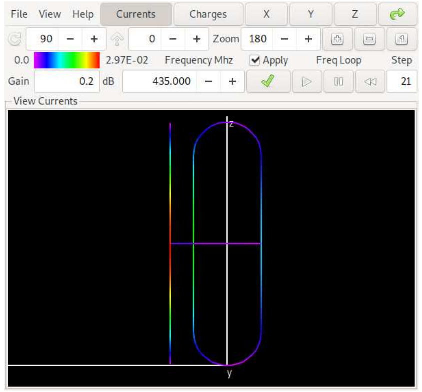


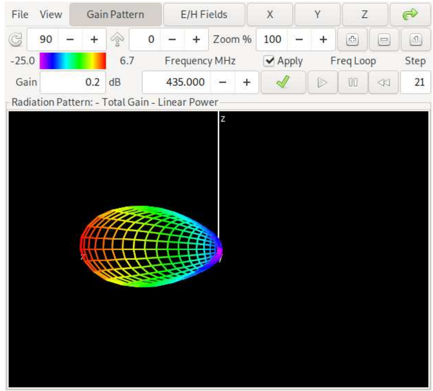
Antenna Lower Pareto Front





Antenna Upper Pareto Front





Software

- PGApack: github.com/schlatterbeck/pgapack
- PGApy: github.com/schlatterbeck/pgapy
- PyNEC: pypi.org/project/PyNEC/
- Antenna-Optimizer: github.com/schlatterbeck/antenna-optimizer
- Plot-Antenna: github.com/schlatterbeck/plot-antenna

Bibliography

[AL95] Edward E. Altshuler and Derek S. Linden. Process for the design of antennas using genetic algorithms. US Patent US5719794A, July 1995.

[Lin97] Derek S. Linden. *Automated Design and Optimization of Wire Antennas Using Genetic Algorithms*. Dissertation, Massachusetts Institute of Technology, September 1997.

[PSL05] Kenneth V. Price, Rainer M. Storn, and Jouni A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization*. Springer, Berlin, Heidelberg, 2005.

[Sch23] Maximilian Schlatterbeck. Actuation waveform design for piezo electric inkjet applications using an evolutionary algorithm. Bachelor thesis, FH Technikum Wien, September 2023.

[SP97] Rainer Storn and Kenneth Price. Differential

evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, December 1997.

